A New Block Exact Affine Projection Algorithm

Felix Albu
Department of Telecommunications
“Politehnica” University of Bucharest
Bucharest, Romania
felix_albu@ieee.org

H.K. Kwan
Department of Electrical and Computer Engineering
University of Windsor
Windsor, Canada
kwan1@uwindsor.ca

Abstract—A block affine projection algorithm that it is mathematically equivalent to a recently proposed Gauss-Seidel Pseudo Affine Projection (GS-PAP) algorithm is proposed. A partitioning method is applied to the original sample-by-sample algorithm. It is shown that the derived algorithm has better convergence, tracking abilities and much reduced complexity than the NLMS algorithm. Its application in an acoustic echo cancellation system is investigated.

I. INTRODUCTION

Adaptive filtering is essential in application such as system identification, acoustic echo cancellation, channel equalization, speech coding. In acoustic echo cancellation systems an adaptive filter algorithm is used to reduce the echo. The well-known normalized LMS (NLMS) algorithm has been widely used but it has slow asymptotic convergence. The affine projection algorithm (APA) [1] can be considered as a generalization of the NLMS algorithm. This algorithm provides a much improved convergence speed compared to stochastic gradient descent algorithms. Its performance rivals the recursive-least square algorithms in many situations, but its complexity is high [1]. Its fast version proposed in [2], when implemented with an embedded FRLS (Fast Recursive Least Squares) algorithm suffers from numerical instability. In [3] we proposed a simpler and sub-optimal implementation of the AP algorithm based on Gauss-Seidel method called the Gauss-Seidel Pseudo Affine Projection (GS-PAP) algorithm. This algorithm is shortly presented in Section 2. In order to further reduce the numerical complexity of the affine projection algorithms, several subband and block exact variants were proposed [6-8]. In Section 3 a novel block exact version of the GS-PAP, called Fast Block Exact GS-PAP (FBEAGS-PAP) is proposed. The results of simulations for echo cancellation systems and the computational complexity of the algorithm are evaluated in section 4. Section 5 concludes this work.

II. GS-PAP ALGORITHM

We’ll use most of the notations from [3]. Let take the following notations: \(x(n)\) is the input signal and \(y(n)\) is the desired output signal. \(e(n)\) is the output error, \(\bar{e}(n)\) is the normalized error, \(X(n)=[x(n),...,x(n-L+1)]^T\), where \(L\) is the filter length. \(R(n)\) is the auto-correlation matrix of the signal, \(\xi(n)=[x(n),...,x(n-N+1)]^T\), where \(N\) is the affine projection order. \(\delta(n)\) is a regularization factor that prevents the input auto-correlation matrix from becoming ill-conditioned. The initial value of the regularization parameter is computed as 1% of the maximum possible value of the \(R(n)\). \(U(n)=[u(n),...,u(n-L+1)]^T\) is the approximated de-correlation vector. \(b\) is an \(N\) vector with only one non-zero element, which is unity at the top. \(H(n)=[h_1(n),...,h_N(n)]^T\) is the filter coefficients vector. \(P\) is an \(N\) length vector and \(P_i, i=0\) to \(N-1\) is the \(i\)th element of it. Below are the equations of the derived Gauss-Seidel Pseudo Affine Projection algorithm (GS-PAP) [3]:

\[
\begin{align*}
X(-1)&=0, \ R(-1)=\delta(-1)I, \ P(-1)=b/\delta(-1), \\
U(-1)&=0, \ H(-1)=0 \\
\text{At each sample } n \geq 0, \\
R(n) &= R(n-1) + \xi(n)\xi^T(n) - \delta(n-L)\xi^T(n-L) + \delta(1)I \\
R(n)P(n) &= b \rightarrow \text{solved with Gauss-Seidel method (3)} \\
e(n) &= y(n) - X^T(n)H(n-1) \\
U(n) &= \frac{1}{P_0(n)} \sum_{i=0}^{N-1} P_i(n)X(n-i) \\
\bar{e}(n) &= \frac{\mu}{U^T(n)U(n) + \delta(n)} e(n) \\
H(n) &= H(n-1) + U(n)\bar{e}(n)
\end{align*}
\]

\(^1\)The author is now with Aristotle University of Thessaloniki, Greece
We solve \( \mathbf{R}(n)\mathbf{P}(n) = \mathbf{b} \) with the Gauss-Seidel method [2]. The Gauss-Seidel method is guaranteed to converge because the matrix \( \mathbf{R}(n) \) is symmetric and positive definite. An advantage of the GS-PAP algorithm is that it provides the filter coefficients unlike the other FAP algorithms that compute only auxiliary coefficients [2], [6-7]. Therefore, it can be applied for other system identification applications. The acoustic echo cancellation systems should have a reliable and prompt double-talk detector, in order to avoid the divergence of the adaptive algorithm. Also, in noisy conditions or near-end disturbances the regularization is a necessary part of the affine projection algorithms [4-5]. We used the following regularization that proved efficient in our simulations for the usual values of the projection order.

\[
\delta(n) = \max\{L \cdot \|x(n)\|^2 / 10, L \cdot \|y(n)\|^2, \delta(-1)\}
\]

where the time-averaged powers are obtained by IIR filtering.

### III. FBEGS-PAP ALGORITHM

Next, we’ll derive a novel algorithm, reducing the complexity of GS-PAP by using the same idea of deriving Modified Fast Exact NLMS (MFENLMS) algorithm from the NLMS algorithm [9]. It can be seen that the equations 6 and 7 lead to a similar update equation of the NLMS algorithm. We denote by \( M \) the block size. We partition the vectors \( \mathbf{X}(n) \), \( \mathbf{U}(n) \) and \( \mathbf{H}(n) \) according to:

\[
\mathbf{X}^T(n) = \left[ \begin{array}{c} \mathbf{X}_a^T(n) \mathbf{X}_b^T(n) \mathbf{x}(n-2M) \end{array} \right]
\]

\[
\mathbf{U}^T(n) = \left[ \begin{array}{c} \mathbf{U}_a^T(n) \mathbf{U}_b^T(n) \mathbf{u}(n-2M) \end{array} \right]
\]

\[
\mathbf{H}^T(n) = \left[ \begin{array}{c} \mathbf{H}_a^T(n) \mathbf{H}_b^T(n) \end{array} \right]
\]

with

\[
\mathbf{X}_a^T(n) = \left[ \begin{array}{c} x(n), \ldots, x(n-2M+1) \end{array} \right]
\]

\[
\mathbf{U}_a^T(n) = \left[ \begin{array}{c} u(n), \ldots, u(n-2M+1) \end{array} \right]
\]

\[
\mathbf{H}_a^T(n) = \left[ \begin{array}{c} h_1(n), \ldots, h_{2M}(n) \end{array} \right]
\]

\[
\mathbf{X}_b^T(n) = \left[ \begin{array}{c} x(n-2M), \ldots, x(n-L+1) \end{array} \right]
\]

\[
\mathbf{U}_b^T(n) = \left[ \begin{array}{c} u(n-2M), \ldots, u(n-L+1) \end{array} \right]
\]

\[
\mathbf{H}_b^T(n) = \left[ \begin{array}{c} h_{2M+1}(n), \ldots, h_L(n) \end{array} \right]
\]

The equations (4-7) of the GS-PAP algorithm can be partitioned into an “updating” part and a “fixed” filtering part as in [9]. Computation of the “updating” part with a lower order of complexity is performed at each recursion step, whereas the “fixed” part can be computed from \( M \) recursion steps. Therefore, the non-uniformly distribution of the arithmetic operations within a block of \( M \) samples, encountered in other block algorithms ([7-8], [10]) can be alleviated.

The equation (7) is repeatedly substituted in equation (4) taking into account the previous partitions. The following equations are obtained for \( i = n \ldots n-M+1 \):

\[
\hat{y}_a(i) = \mathbf{X}_a^T(i) \mathbf{H}_a^T(i) + \sum_{j=1}^{M+n} \sigma_j(i-2M) \overline{v}(i-j)
\]

\[
\sigma_j(i-2M) = \mathbf{U}_b^T(i-2M-j) \mathbf{X}_b(i-2M)
\]

\[
\hat{y}_b(n-M) = \mathbf{U}_b^T(i-2M) \mathbf{H}_b(n-M)
\]

\[
\mathbf{H}_a(i) = \mathbf{H}_a(i-1) + \mathbf{X}_a(i-1) \overline{v}(i-1)
\]

\[
\mathbf{H}_b(n) = \mathbf{H}_b(n-M) + \sum_{j=1}^{M} \mathbf{U}_b(n-2M-j) \overline{v}(n-j)
\]

\[
e(i) = y(i) - \hat{y}_a(i) - \hat{y}_b(n-M)
\]

The Fast Block Exact GS-PAP (FBEGS-PAP) algorithm includes the equations 2-3, 5-6, 18-23. All equations excepting equations 20 and 22 are performed every step. Their complexity is small because usually \( M \) and \( N \) are much smaller than the filter length \( L \). The “filtering part” (equations 20 and 22) depends on the data from the last block. This part involves the computation of \( M \) successive outputs of a fixed coefficient filter. Several fast FIR filtering procedures that reduce significantly the number of operations exists (e.g. based on linear or circular convolution). We used the efficient FIR filtering architecture presented in Fig. 2 of [11] for cases when the block size is a power of two \( (M = 2^k) \). The numerical complexities of the algorithms are measured by counting the number of multiplications and divisions per recursion. Therefore we have:

\[
C_{\text{NLMS}} = 2L + 4
\]

\[
C_{\text{MFENLMS}} = 10M + 1 + [2(3/4)^{k} (L-2M)]
\]

\[
C_{\text{FBEGS-PAP}} = 10M + 2 + [2(3/4)^{k} (L-2M)] + N^2 + 4N
\]

\[
C_{\text{GS-PAP}} = 2L + N^2 + 3N + 5
\]

The increase in complexity of FBEGS-PAP algorithm in comparison with MFENLMS is rather small for the usual values of \( N \) and doesn’t depend on \( L \) or \( M \). The proper choice of the block size has a crucial importance in reducing the number of multiplies and divisions. As shown in [9], there is an “optimal” block size value of \( M \) that minimizes its complexity if the projection order is fixed. The FBEGS-PAP algorithm has a more uniform distribution of the operations within the block consisting of \( M \) samples and compares favorably with other such algorithms from complexity point of view. Different block sizes can be used for each fast FIR procedure, but the practical implementation is more complicated.
IV. SIMULATIONS

We tested the convergence and tracking abilities of the FBEGS-PAP algorithm using speech as excitation signals in an acoustic echo cancellation example. As expected, because it is mathematically equivalent with GS-PAP, it inherits its performances. An example of the superior convergence and tracking abilities of the FBEGS-PAP algorithm to the NLMS when using speech excitation signals is shown in Fig. 1. In our simulations we used \( L = 256, N = 10, M = 8 \). The impulse response of the measured car cabin impulse response was truncated to 259 coefficients, so that the theoretical minimum misalignment was calculated to be \(-49.06\, \text{dB}\). The tracking abilities of the algorithms were investigated by a sudden change in the sign of the measured echo path coefficients after 25,000 samples.

![Figure 1](image1.png)

**Figure 1.** The misalignment of FBEGS-PAP and NLMS algorithms for speech excitation under a sudden change in echo path \((L = 256, N = 10, M = 8, \mu = 1)\).

The proper choice of the block size has a crucial importance in reducing the number of multiplies and divisions (see Fig. 2). There is an “optimal” block size value of \( M \) in order to minimize the complexity indicated by equation 26.

<table>
<thead>
<tr>
<th>ALGORITHM</th>
<th>( L = 256, M = 8 )</th>
<th>( L = 1024, N = 10 )</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>( N = 10 )</td>
<td>( N = 4 )</td>
</tr>
<tr>
<td>FBEGS-PAP</td>
<td>425 (335)</td>
<td>918 (828)</td>
</tr>
<tr>
<td>GS-PAP [2]</td>
<td>647 (557)</td>
<td>2183 (2093)</td>
</tr>
<tr>
<td>MFENLMS [7]</td>
<td>284 (284)</td>
<td>777 (2052)</td>
</tr>
<tr>
<td>NLMS</td>
<td>516</td>
<td>2052</td>
</tr>
</tbody>
</table>

For example (see Fig. 3), if \( L = 8192 \), the optimum block size is 128 (i.e. \( k = 7, M = 2^k \)). It can be seen that the value of the block size that minimize the number of multiplies and divisions is higher for higher filter lengths (e.g. if \( N = 10 \), the “optimal” value of \( M \) is 8 for \( L = 256 \), 32 for \( L = 1024 \) etc.). This dependence of “optimal” \( M \) with the filter length is represented in Fig. 3. Fig. 4 shows that the FBEGS-PAP algorithm has a good behavior in a double-talk scenario.

![Figure 2](image2.png)

**Figure 2.** The number of multiplies and divisions of FBEGS-PAP as a function of the block size \( M = 2^k \) (\( L=8192, N=10 \)).
Therefore, if a certain delay is acceptable, the FBEGS-PAP algorithm is an attractive algorithm for practical real-time implementations of AEC systems although it needs more memory than GS-PAP or other fast affine projection algorithms.

V. CONCLUSIONS
We've proposed an efficient implementation of the GS-PAP algorithm. It was proved that the FBEGS-PAP algorithm has significantly better performances and reduced complexity in comparison with NLMS. Its stability and uniform distribution of operations recommends it for practical acoustic echo cancellation systems.

REFERENCES

The matlab code for the proposed algorithm can be obtained from http://falbu.50webs.com/gs.html

The reference for the paper is: